

AD-A125 682

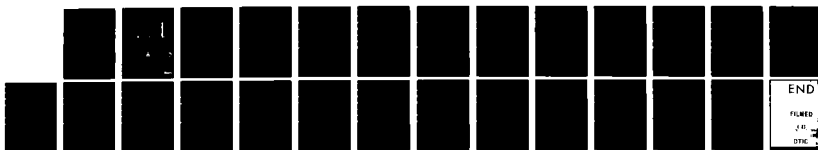
ON THE RECOGNITION OF PROPERTIES OF THREE-DIMENSIONAL
PICTURES(U) MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE
CENTER A NAKANURA JUL 82 CSC-TR-1192 AFOSR-TR-83-0071
AFOSR-77-3271

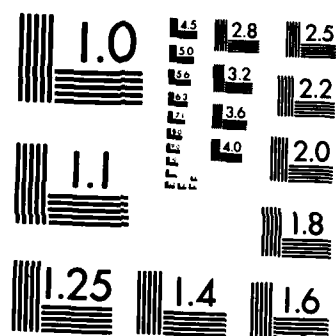
1/1

UNCLASSIFIED

F/G 9/4

NL

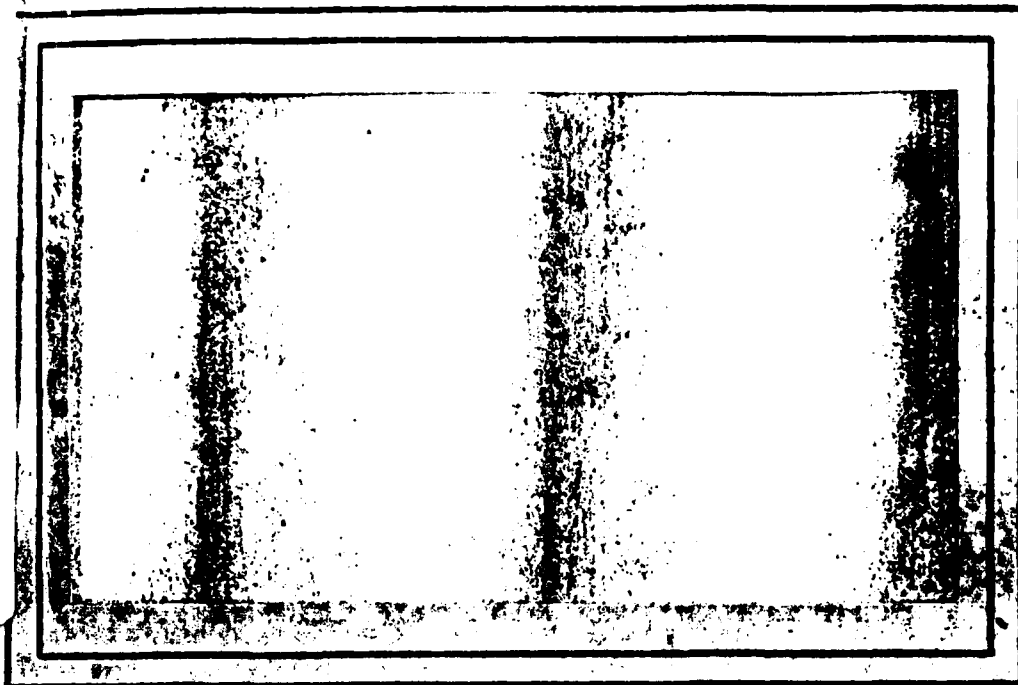




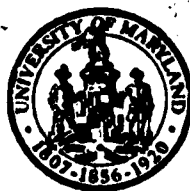
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

4

AD A125602



COMPUTER SCIENCE
TECHNICAL REPORT SERIES



DTIC
ELECTE

MAR 14 1983

A

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND

20742

Approved for public release;
distribution unlimited.

ITG FILE COPY

88 03 14 016

UNCLASSIFIED

TR-1192
AFOSR-77-3271

July, 1982

ON THE RECOGNITION OF PROPERTIES OF
THREE-DIMENSIONAL PICTURES

Akira Nakamura
Kunio Aizawa
Department of Applied Mathematics
Hiroshima University
Higashi-Hiroshima, 724
Japan

MAR 14 1983

ABSTRACT

A

It is shown that a one-way 3D parallel/sequential acceptor cannot accept the class of 3D binary arrays in which the set of 1's is connected, unlike the situation in the 2D case; but it can do so if multiple passes are allowed.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTES ON THE RECOGNITION OF PROPERTIES OF
THREE-DIMENSIONAL PICTURES
MAITHEM. J.
Chief, Technical Information Division

The support of the U.S. Air Force Office of Scientific Research under Grant AFOSR-77-3271 is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper. The first author, Akira Nakamura, would like to thank Professor Azriel Rosenfeld for arranging a very enjoyable stay at the Computer Science Center, University of Maryland, July 1982, and for valuable comments.

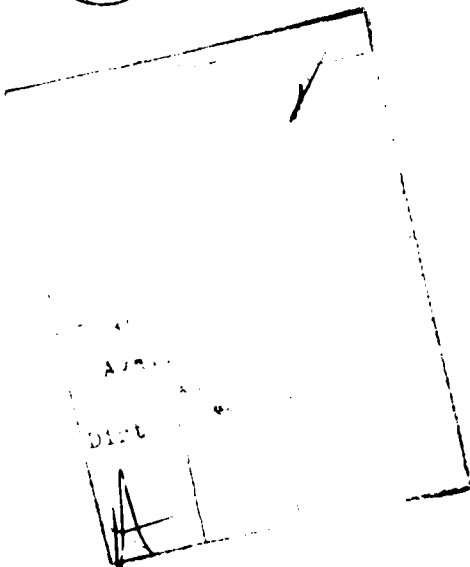
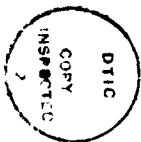
0. Introduction

In recent years, there have arisen many requirements for three-dimensional (3D) data processing with advances in computer tomography (CT). Some topological properties of 3D digital pictures have been discussed in a series of papers by Rosenfeld and Morgenthaler [1]-[5], as well as others. A 3D picture can be represented by a 3D array of volume elements (voxels for short). In a binary-valued 3D picture (each voxel is 0 or 1), it is easy to define connectedness, and objects and cavities are then defined as the equivalence classes of the connectedness relation. These correspond to two-dimensional (2D) objects and holes, respectively. Moreover, in the 3D case, there also exist 3D holes whose properties are quite different from those of 2D holes.

In this paper, we consider the problem of recognition of the above-mentioned properties of 3D pictures. First, we propose algorithms which for every (binary) 3D digital picture compute the numbers of objects, cavities and holes. These algorithms are performed in one pass; they are 3D versions of the algorithm for 2D digital pictures which was given by Selkow [6]. Since we usually receive a series of 2D pictures as the output of a CT scanner, it is of interest that these algorithms scan such output plane by plane, from the top to the bottom, and stop when all of them have been scanned. However, the algorithms violate the condition of finiteness of the symbols used. From the point of view of the theory of languages and their acceptance, this condition of finiteness

is a fundamental one. In the second part of this paper, we discuss this problem and show that any one-way 3D parallel/sequential acceptor cannot accept the set of 3D pictures such that the set of 1's is connected. This is interesting because it is known that a one-way 2D parallel/sequential acceptor can do this acceptance for the 2D case (e.g., see Theorem 7.4.6 in [7]). In the third part of this paper, we extend the concept of a 3D parallel/sequential acceptor to allow multiple passes. This concept of multi-pass is the same as that introduced in [8]. Then, making use of a shrinking technique used in [9] we show that a multi-pass one-way 3D parallel/sequential acceptor can accept 3D connectedness.

We assume that the readers are familiar with the basic concepts of picture languages.



1. Preliminaries

Let Σ be a 3D array of lattice points, which we assume to be $\ell \times m \times n$, i.e., $\Sigma = \{(i,j,k) \mid 1 \leq i \leq \ell, 1 \leq j \leq m, 1 \leq k \leq n\}$. A 3D digital picture f is a mapping from Σ to $\{0,1\}$, i.e., $f: \Sigma \rightarrow \{0,1\}$. Each point (i,j,k) is called a voxel. Here, we assume that any 3D digital picture is surrounded by the special symbol # (called the blank symbol). That is, let B be the set $\{(i,j,k) \mid (i=0 \ \& \ 0 \leq j \leq m+1 \ \& \ 0 \leq k \leq n+1) \vee (i=\ell+1 \ \& \ 0 \leq j \leq m+1 \ \& \ 0 \leq k \leq n+1) \vee (0 \leq i \leq \ell+1 \ \& \ j=0 \ \& \ 0 \leq k \leq n+1) \vee (0 \leq i \leq \ell+1 \ \& \ j=m+1 \ \& \ 0 \leq k \leq n+1) \vee (0 \leq i \leq \ell+1 \ \& \ 0 \leq j \leq m+1 \ \& \ k=0) \vee (0 \leq i \leq \ell+1 \ \& \ 0 \leq j \leq m+1 \ \& \ k=n+1)\}$. Then every lattice point of B is occupied by the symbol #. The set B is called the border of Σ . Usually, the subset of 1's of Σ , i.e., the set $\{(i,j,k) \mid f(i,j,k)=1\}$, is called S , and its complement is called \bar{S} . For every pair of points $X=(x_1, x_2, x_3)$ and $Y=(y_1, y_2, y_3)$, X and Y are 6-adjacent if $|x_1-y_1| + |x_2-y_2| + |x_3-y_3| = 1$; X and Y are 26-adjacent if $\max(|x_1-y_1|, |x_2-y_2|, |x_3-y_3|) = 1$. If points P and Q are 6-adjacent (26-adjacent), then P is called a 6-neighbor (26-neighbor) of Q . To avoid ambiguous situations we assume that opposite types of adjacency are used for S and \bar{S} . A 6-path (26-path) π is a sequence of points, $\pi = p_0, p_1, \dots, p_m$, where, for all i such that $1 \leq i \leq m$, p_i is a 6-neighbor (26-neighbor) of p_{i-1} . Any two points P, Q of S are called connected in S if there exists a path $P=p_0, \dots, p_m=Q$ from P to Q , where $p_i \in S$. Evidently, "connected" is an equivalence relation. This relation partitions S into equivalence classes. These classes are called the connected components of S . In the same way,

we may define connectedness in \bar{S} and the connected components of \bar{S} . A connected component of S is called an object of S . Clearly, exactly one component of \bar{S} contains the #'s. This component is called the background of S ; all other components of \bar{S} are called cavities of S .

Even in ordinary topology it is difficult to characterize holes. A hole may be thought of as a property of a boundary surface which makes it topologically equivalent to a torus. In another approach, an object is defined to have no hole if every simple closed curve in the object is continuously deformable within the object to a single point. We see from these remarks that the concept of a hole is different from those of objects and cavities; we cannot point to or label the points which constitute a hole. Indeed, the points of the objects and cavities cover the space, but a hole is a property of those collections of points. Thus, when considering an object (and its cavities) we shall here try only to understand what is meant by the number of holes in the object, and not what is meant by a hole.

The genus $G(S)$ of a set S in a 3D digital picture is defined as the number of objects in S ($O(S)$) plus the number of cavities in S ($C(S)$) minus the number of holes in S ($H(S)$). As already mentioned the definition of holes is not simple, and in particular holes cannot be labelled to facilitate counting them. But since this can be done with objects and cavities, the definition of the genus defines the number of holes in S , and conversely.

In [4], Morgenthaler has given a method of computing $G(S)$ directly from the local patterns in S :

1) When 26-adjacency is used for S ,

$$G_{26}(S) = \phi_1 - \phi_2 + \phi_3 - \phi_4 + \phi_5 - \phi_6 + \phi_7 - \phi_8,$$

where

$$\phi_1 = \#(2),$$

$$\phi_2 = \#(3) + \#(4) + \#(5),$$

$$\phi_3 = \#(6) + \#(7) + \#(8),$$

$$\phi_4 = \#(9) + \#(10) + \#(11) + \#(12) + \#(13) + \#(14),$$

$$\phi_5 = \#(15) + \#(16) + \#(17),$$

$$\phi_6 = \#(18) + \#(19) + \#(20),$$

$$\phi_7 = \#(21),$$

$$\phi_8 = \#(22),$$

and by $\#(n)$ we mean the number of times the configuration n of Fig. 1 occurs in the picture S (in all orientations).

2) When 6-adjacency is used for S ,

$$G_6(S) = \psi_1 - \psi_2 + \psi_3 - \psi_4,$$

where

$$\psi_1 = \#(2),$$

$$\psi_2 = \#(3),$$

$$\psi_3 = \#(9),$$

$$\psi_4 = \#(22),$$

The 22 patterns used in these definitions are drawn in Figure 1.

Morgenthaler has also shown that $G_{26}(\bar{S}) - G_6(S) = 1$ and $G_6(\bar{S}) - G_{26}(S) = 1$.

2. The number of objects, cavities and holes

In this section, we consider algorithms that compute the number of objects, cavities, and holes in a 3D digital picture.

A scanner is an $\ell \times m$ array of finite-state automata

$$\sigma_{11} \sigma_{12} \cdots \sigma_{1m}$$

$$\sigma_{21} \sigma_{22} \cdots \sigma_{2m}$$

$$\sigma_{\ell 1} \sigma_{\ell 2} \cdots \sigma_{\ell m}$$

Each automaton σ_{ij} is defined by a 7-tuple $\langle Q_{ij}, \delta_{ij}, \alpha_{ij}, a_{ij}, \lambda_{ij}, \beta_{ij}, b_{ij} \rangle$, where Q_{ij} , the set of states, is a set of integers, and δ_{ij} , the next state function, has the following form:

$$\delta_{ij}: \prod_{k=1}^{a_{ij}} Q_{\alpha_{ij}(k)} \times \{0,1\} \rightarrow Q_{ij}$$

where a_{ij} is an integer, and $\alpha_{ij}: \{1, \dots, a_{ij}\} \rightarrow I \times I$ is a one-to-one function which enumerates the next state neighborhood of

σ_{ij} . λ_{ij} , the output function, has the following form:

$$\lambda_{ij}: \prod_{k=1}^{b_{ij}} Q_{\beta_{ij}(k)} \times \{0,1\} \rightarrow I,$$

where b_{ij} is an integer, and $\beta_{ij}: \{1, \dots, b_{ij}\} \rightarrow I \times I$ is a one-to-one function which enumerates the output neighborhood of σ_{ij} .

We will use $Q_{ij}(t)$ to represent the state of σ_{ij} at time t . It is assumed that the scanner advances by one plane in each unit of time and that it scans the first plane at time $t=1$. Thus, the input to scanner element σ_{ij} at time t is $f(i,j,t)$.

Note that Q_{ij} is not a finite set and also that the neighborhood of σ_{ij} is not fixed.

The counter C monitors the output of each element of the scanner; thus $C(t) = C(t-1) + \sum_{i=1}^{\ell} \sum_{j=1}^m \lambda_{ij}(t)$.

Now, we shall describe algorithms for computing the numbers of objects, cavities, and holes.

1: Objects

The set of states of scanner element σ_{ij} is $Q_{ij} = \{x | x \text{ is an integer and } |x| \leq (i+j-1)(i+j-2)/2+j\}$. Each automaton σ_{ij} starts in state 0 and remains in that state as long as 0's are scanned. When a voxel containing a 1 is reached, σ_{ij} will assume state $(i+j-1)(i+j-2)/2+j$. As σ_{ij} tracks a string of 1's, an extension of the component of S is sought, i.e., two automata which are actively tracking 1's are tracking the same object if they are spatially neighbors or if they are in the same state. All automata which are tracking the same object assume the same state (the state of the automaton having the smallest state). An automaton which passes a lower border of an object and has been in a state k enters the state $-k$ for one period. The next time it goes directly to state 0 unless a 1 is encountered. If σ_{ij} enters the state $-((i+j-1)(i+j-2)/2+j)$ and no others are in the state $(i+j-1)(i+j-2)/2+j$, then σ_{ij} will output 1, i.e., one object has been scanned. The precise definitions of δ_{ij} and λ_{ij} are represented as follows:

i) When 6-adjacency is used for S,

$$\delta_{ij}(Q_{ij}(t-1), f(t, i, j))$$

$$= \begin{cases} ((i+j-1)(i+j-2)/2+j) \cdot f(t, i, j) & \text{if } Q_{ij}(t-1) \leq 0 \\ \text{LINK}(Q_{ij}(t-1)) & \text{if } Q_{ij}(t-1) > 0 \wedge f(t, i, j) = 1 \\ -\text{LINK}(Q_{ij}(t-1)) & \text{if } Q_{ij}(t-1) > 0 \wedge f(t, i, j) = 0, \end{cases}$$

$$\lambda_{ij} = \begin{cases} 1 & \text{if } Q_{ij}(t) = -((i+j-1)(i+j-2)/2+j) \text{ for all } x, y \text{ such that} \\ & -(x+y-1)(x+y-2)/2+y > (i+j-1)(i+j-2)/2+j, \\ 0 & \text{otherwise,} \end{cases} \quad Q_{xy}(t) \neq -Q_{ij}(t),$$

where

LINK(g)

$$= \min_{1 \leq k \leq n} 2 \{ (x_k + y_k - 1)(x_k + y_k - 2)/2 + y_k \mid (\exists x_1) \dots (\exists x_{k-1}) (\exists y_1) \dots (\exists y_{k-1}) (Q_{x_1 y_1}(t-1) = g) \wedge (\forall j) (2 \leq j \leq k) ((Q_{x_j y_j}(t-1) > 0) \wedge ((x_j, y_j) \text{ and } (x_{j-1}, y_{j-1}) \text{ are 6-adjacent}) \vee (Q_{x_j y_j}(t-1) = Q_{x_{j-1} y_{j-1}}(t-1))) \}.$$

ii) When 26-adjacency is used for S,

$$\delta_{ij}(Q_{ij}(t-1), f(t, i, j))$$

$$= \begin{cases} 0 & \text{if } Q_{ij}(t-1) \leq 0 \wedge f(t, i, j) = 0 \\ \text{LINK}((i+j-1)(i+j-2)/2+j) & \text{if } f(t, i, j) = 1 \\ -\text{LINK}((i+j-1)(i+j-2)/2+j) & \text{if } Q_{ij}(t-1) > 0 \wedge f(t, i, j) = 0 \end{cases}$$

λ_{ij} is the same as in the case (1),

where

LINK(g)

$$= \min_{1 \leq k \leq n} 2^{\{(x_k+y_k-1)(x_k+y_k-2)/2+y_k \mid (\exists x_1) \dots (\exists x_{k-1}) (\exists y_1) \dots (\exists y_{k-1}) ((x_1+y_1-1) \cdot (x_1+y_1-2)/2+y_1$$

$$= g) \wedge (\forall j) (2 \leq j \leq k) ((Q_{x_j, y_j}(t-1) > 0) \wedge ((x_j, y_j) \text{ and } (x_{j-1}, y_{j-1}) \text{ are } 26\text{-adjacent}) \vee (Q_{x_j, y_j}(t-1) = Q_{x_{j-1}, y_{j-1}}(t-1)))\}.$$

2: Cavities

Since all components of \bar{S} except the background component are cavities of S , the algorithm for counting objects of S can be used for counting cavities of S by interchanging the roles of 1 and 0. In this case, the initial value of the counter C must be to remove the background component from the set of cavities of S .

3: Genus

For any S , every $2 \times 2 \times 2$ local pattern in Fig. 1 is easily counted by our computational model. Thus the algorithm computing the genus of S is easily defined.

4: Holes

Finally from algorithms 1-3, we can construct the algorithm for counting the number of holes in S , since $H(S) = O(S) + C(S) - G(S)$.

3. Unacceptability of the connectedness of a 3D picture by a deterministic one-way PSA

In Section 2, we have given algorithms which for every 3D digital picture count the number of objects, cavities, and holes. However, as noted in Section 2 the σ_{ij} used in the algorithms needs a set of states which is not finite. Further, its neighborhood is not fixed. In language theory, finiteness is required in the automaton used. From the point of view of language theory, we prove here the unacceptability of the connectedness of 3D pictures by deterministic one-way parallel/sequential acceptors. Parallel/sequential acceptors of 2D pictures are well-known (e.g., see [7]). It is also known that a deterministic one-way non-writing parallel/sequential acceptor can determine whether or not the set of x's in a rectangular array Σ of x's and y's is connected. In contrast with this result, our new theorem for 3D seems to be interesting.

A parallel/sequential acceptor of 3D pictures (for brevity: 3DPSA) is analogous to that of 2D pictures. It is defined as follows: A parallel/sequential acceptor is a 9-tuple $A=(Q, q_0, Q_A, \#, V, \#_t, \#_b, \delta, \mu)$, where

Q is a finite nonempty set of states

$q_0 \in Q$ is the initial state

$Q_A \subseteq Q$ is the set of accepting states,

$\# \in Q$ is the blank symbol,

V is a finite nonempty set of symbols called the tape vocabulary,

$\#_t$ and $\#_b$ are blank symbols in V ,

$\delta: Q^9 \times V \rightarrow 2^{Q \times V}$ is the state transition function and

$\mu: Q \times V \rightarrow 2^{\{-1,0,1\}}$ is the move function.

The operation of A on a 3D array Σ can be described as follows:

A consists of a 2D array of cells

c_{11}	...	c_{1m}
c_{21}	...	c_{2m}
.....		
$c_{\ell 1}$...	$c_{\ell m}$

whose lengths are the numbers of columns and rows of Σ , respectively, together with special "cells"

c_{00}	c_{01}	...	$c_{0 \ m+1}$
c_{10}			$c_{1 \ m+1}$
c_{20}			$c_{2 \ m+1}$
\vdots			\vdots
$c_{\ell 0}$			$c_{\ell m+1}$
$c_{\ell+1 \ 0}$...		$c_{\ell+1 \ m+1}$

that are regarded as permanently in the # symbol. Σ has a row of $\#_t$'s just above its top array and a row of $\#_b$'s just below its bottom array. Initially, A is on the top array of Σ with every cell in the state q_0 . At any given step, each cell c_{ij} reads the symbol v in its position, senses the states q_1, q_2, \dots, q_9 of $c_{i-1, j-1}$, $c_{i-1, j}$, $c_{i-1, j+1}$, $c_{i, j-1}$, $c_{i, j}$, $c_{i, j+1}$, $c_{i+1, j-1}$, $c_{i+1, j}$, $c_{i+1, j+1}$ and can go into any new state q' and write any new symbol v' such that $(q', v') \in \delta(q_1, q_2, \dots, q_9)$. The move function depends only on

the (new) state and symbol read by the distinguished cell c_{11} . $0 \in \mu$ means that A can stay where it is; $1 \in \mu$ means that A can move down, $-1 \in \mu$ means it can move up.

It is required that $\mu(q, \#_t) = 1$ and $\mu(q, \#_b) = -1$ for all q . It is understood that $\#_t$ and $\#_b$ can never be rewritten. If c_{11} ever enters a state in Q_A , we say that A has accepted Σ .

If a PSA does not move up, A is called one-way.

Now, we prove the following theorems. Let C be the set of 3D arrays consisting of 0's and 1's in which the 1's are connected.

Theorem 2.1 C is accepted by a nondeterministic one-way 3DPSA.

Proof: Let us consider isometric 3D array grammars, which are three-dimensional analogs of those in the 2D case (see [7]). Notice that C is generated by an isometric 3D monotonic array grammar G. Here, we can assume that the starting symbol ρ of G appears only once during the applications of rewriting rules.

Now, let A be a nondeterministic one-way 3DPSA working as follows:

- 1) When σ_{ij} reads a symbol, the state changes into one corresponding to a non-terminal symbol of G. This change is done nondeterministically.
- 2) Illegal guessing in 1) causes σ_{ij} to go into a dead state.
- 3) When σ_{ij} goes into the state corresponding to the starting symbol ρ , this state moves to σ_{11} . In this case, A never goes downward until this move is completed.
- 4) For two states ρ , σ_{11} goes into the dead state.

- 5) When A reaches the bottom plane and finishes its transitions, σ_{11} goes into the accepting state if it has memorized the only ρ .

From the above construction, we have the theorem. ||

Theorem 2.2 C is not accepted by any deterministic one-way 3DPSA.

Proof: Let us consider a square array of side length 4ℓ as shown in Fig. 2. The left and right halves are called the L-part and R-part, respectively. Cells of both parts are occupied by 0's and 1's as shown in Fig. 3. That is, every even row and every even column are occupied by 1's, and all other cells by 0. This array is put in some plane of Σ . In the arrays other than this plane, the 1-cells are connected as shown in Fig. 4. Here, n_i is the name of a cell in the L-part and m_j is the name of a cell in the R-part. Thus, the cells n_i and n_{i+1} in the same parentheses are connected in the cells between the top array and pth array, and the same for the m_j and m_{j+1} cells.

Further, the line indicated in Fig. 4 shows that the n_i and m_j cells are connected in the cells under the pth array. This connection is always possible by considering a very large p and sufficiently high Σ .

Now, the number of configurations of states of A at the time it leaves the pth array is $|Q|^{4\ell \times 4\ell}$. Also, the number of connections of cells in the L-part (that is, the number of parenthesized

1-cells) is $2^{\ell^2} 2^{\ell^2-2} 2^{\ell^2-2} \dots 2^{\ell^2-2} = \frac{2^{\ell^2}!}{2^{\ell^2}}$. But we have

$|Q|^{16\ell^2} < \frac{2^{\ell^2}!}{2^{\ell^2}}$ for sufficiently large ℓ . Hence, there are at least two different parenthesizings which yield the same configuration of states of A at the time it leaves the pth array. Let these two be as follows:

... (n_1, n_2) (n_3, n_4) ...

... (m_1, m_2) (m_3, m_4) ...

and

... (n_1, n_3) () ...

... (m_1, m_2) (m_3, m_4) ...

Let Fig. 5 and Fig. 6 represent these two 3D arrays, where the connection relations are in the cells between the (p+1)st and bottom arrays. The lines in Fig. 5 show that all 1-cells are connected. This is possible by considering a suitable connection level.

Suppose that a deterministic one-way 3DPSA A accepts the connected 3D arrays. Then A accepts the connected 3D array shown in Fig. 5. But then, A must also accept the non-connected 3D array shown in Fig. 6, since it gives rise to the same configuration of states of A at the time it leaves the pth array. This is a contradiction. Therefore, we have the theorem. //

4. Acceptability of the connectedness of 3D pictures by a multi-pass one-way PSA

In the previous section, we showed that any deterministic one-way PSA cannot accept the connectedness of 3D pictures. Here, we modify the one-way PSA so it can make repeated passes over a given 3D picture. That is, the new acceptor A acts as follows:

- 1) First, A works as the usual one-way PSA. This is the first pass.
- 2) When A reaches the array below the bottom, A begins to work again from the top array. This is the second pass.
- 3) A repeats the behavior 2).
- 4) If c_{11} ever enters a state in Q_A during the repeated behavior of 3), we say that A accepts Σ .

For this multi-pass 3DPSA, we have the following theorem:

Theorem 3.1 A multi-pass deterministic one-way 3DPSA can accept the connectedness of 3D pictures.

Proof: Let $u(t)=1$ if $t>0$, $u(t)=0$ if $t\leq 0$. We consider 3D pictures consisting of 0's and 1's. In Fig. 7 a,b,c,d,e,f,g,h are the voxels belonging to a $2\times 2\times 2$ window.

Then $F(b)$, the transformed value of b , will depend on the values of the elements belonging to the three planes that meet at B.

For 6-connectivity, we define

$$F(b)=u(u(a+b-1)+u(b+c-1)+u(a+d+c-2))+u(f+b-1)+u(c+g+f-2)+u(a+e+f-2)).$$

Similarly, for 26-connectivity, we define

$$F(b)=u(u(a+b+g-1)+u(b+c+e-1)+u(b+f+d-1)+u(a+c+f-1)+u(b+h-1)).$$

Then the following theorem is well-known [9]: When F is applied repeatedly in parallel, a single object shrinks down to a single 1-voxel in a finite number of steps.

By making use of this result, we can prove the theorem. This is done as follows:

In the first pass of a multi-pass deterministic one-way PSA A , the values of a, c, d, e, f, g, h , are written at b as the output of A . This is done for all voxels.

In the second pass, A writes the result obtained by the first application of the function F to Σ . Generally, in the $(n+1)$ st pass A writes the result obtained by the n th application of F to Σ .

When A recognizes a single 1-voxel, c_{11} enters the accepting state. Therefore, we have the theorem. //

By a similar technique, A can accept Σ such that it has a cavity. Furthermore, by making use of a counter (bounded to the number of voxels in Σ) a multi-pass deterministic one-way PSA A can count the number of objects and cavities in Σ . Thus, we know from the equation given earlier that a multi-pass deterministic one-way PSA can count the number of holes in Σ .

References

1. A. Rosenfeld, Three-dimensional digital topology, TR-936, Computer Science Center, University of Maryland, College Park, MD, September 1980.
2. A. Rosenfeld and D. G. Morgenthaler, Some properties of digital curves and surfaces, TR-942, Computer Science Center, University of Maryland, College Park, MD, September 1980.
3. D. G. Morgenthaler and A. Rosenfeld, Surfaces in three-dimensional digital images, TR-940, Computer Science Center, University of Maryland, College Park, MD, September 1980.
4. D. G. Morgenthaler, Three-dimensional digital topology: The genus, TR-980, Computer Science Center, University of Maryland, College Park, MD, November 1980.
5. D. G. Morgenthaler, Three-dimensional simple points: Serial erosion, parallel thinning, and skeletonization, TR-1005, Computer Science Center, University of Maryland, College Park, MD, February 1981.
6. S. M. Selkow, One-pass complexity of digital picture properties, J.ACM 19, 283-295, 1972.
7. A. Rosenfeld, Picture Languages, Academic Press, New York, 1979.
8. K. Inoue and A. Nakamura, Two-dimensional multipass on-line tessellation acceptors, Information and Control, 41, 305-323, 1979.
9. C. Arcelli and S. Levialdi, Parallel shrinking in three dimensions, CGIP 1, 21-30, 1972.

(All zeros)



2 (=21)



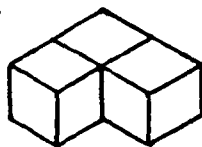
3 (=20)



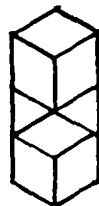
4 (=19)



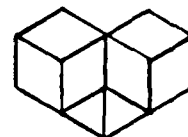
5 (=18)



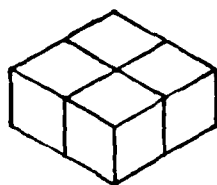
6 (=17)



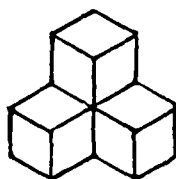
7 (=16)



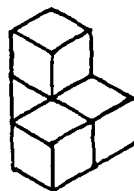
8 (=15)



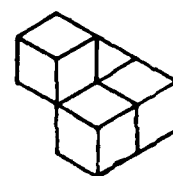
9 (=9)



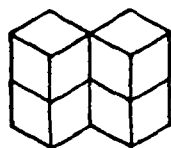
10 (=10)



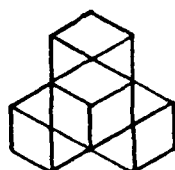
11 (=11)



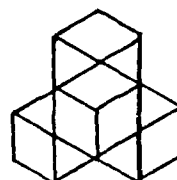
12 (=12)



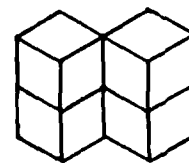
13 (=13)



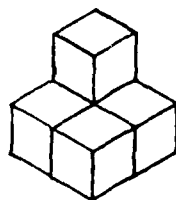
14 (=14)



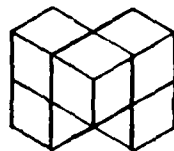
15 (=8)



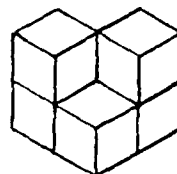
16 (=7)



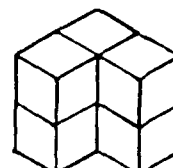
17 (=6)



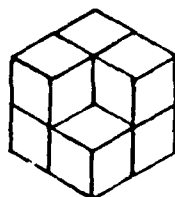
18 (=5)



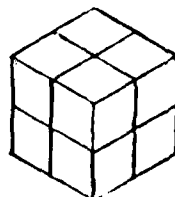
19 (=4)



20 (=3)



21 (=2)



22 (=1)

Fig. 1

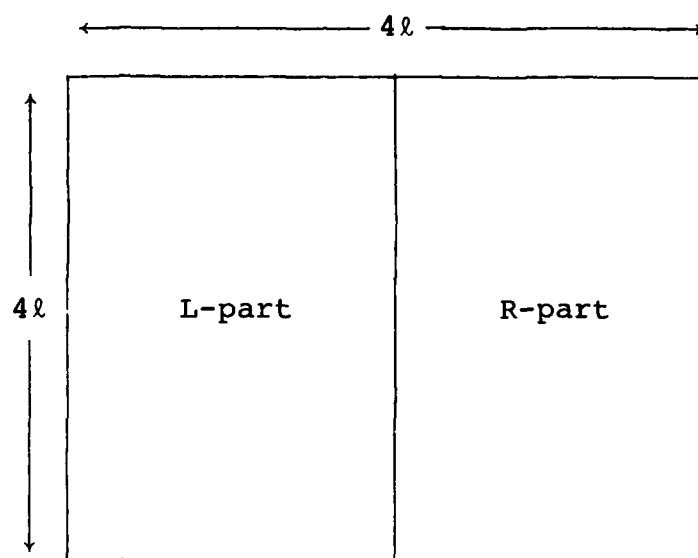


Fig. 2

	1	2	3	4	2ℓ
1	0	0	0	0	0 0
2	0	1	0	1	0 1
3	0	0	0	0	0 0
2ℓ	0	1	0	1	0 1
$2\ell+1$	0	0	0	0	0 0
	0	1	0	1	0 1
	0	0	0	0	0 0
4ℓ	0	1	0	1	0 1

Fig. 3

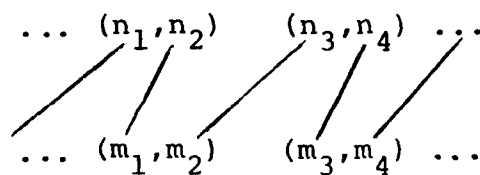


Fig. 4

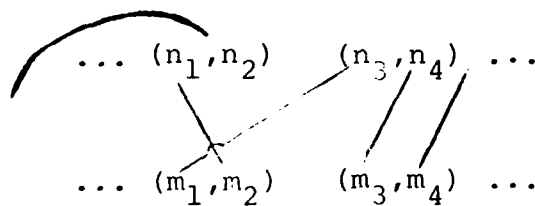


Fig. 5

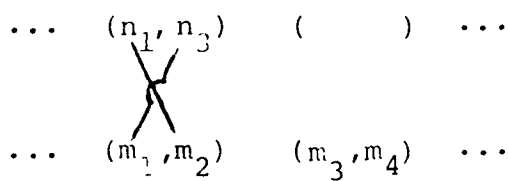


Fig. 6

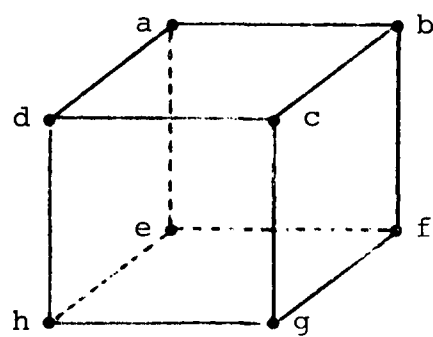


Fig. 7

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83 - 0071	2. GOVT ACCESSION NO. AD-A125602	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ON THE RECOGNITION OF PROPERTIES OF THREE-DIMENSIONAL PICTURES		5. TYPE OF REPORT & PERIOD COVERED Technical
7. AUTHOR(s) Akira Nakamura		6. PERFORMING ORG. REPORT NUMBER TR-1192
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory Computer Science Center University of Maryland College Park, MD 20742		8. CONTRACT OR GRANT NUMBER(s) AFOSR-77-3271
11. CONTROLLING OFFICE NAME AND ADDRESS Math & Info. Sciences, AFOSR/NM Bolling AFB Washington, DC 20332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A2
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE July 1982
		13. NUMBER OF PAGES 23
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Image processing Pattern recognition 3D images Automata Connectedness		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) It is shown that a one-way 3D parallel/sequential acceptor cannot accept the class of 3D binary arrays in which the set of 1's is connected, unlike the situation in the 2D case; but it can do so if multiple passes are allowed.		

4-8
DTI